COBOL Compiles on the Cloud

Richard Ralston
Humana Inc.

Share – San Francisco
Session 12918
February 6, 2013 – 15:00
Its All About Saving Money

- Mainframe cost containment, software and hardware forces us to continually look for more cost effective solutions
- The mainframe is here to stay
  - It’s the heart of IT
  - Major business processes all have the majority of their data in mainframe storage
- We’ve already picked the low hanging fruit
- We’ve provided a slower cost growth
  - Sub-Capacity pricing, audited and verified every month
  - Hardware upgrades
    - Technology dividend on software – VWLC/AWLC pricing
    - Better lease/purchase terms
  - Software cost offload – zIIPs, zAAPs
  - LPAR Defined Capacity/Group Capacity Limits
  - Negotiated better pricing on 3rd party software
  - Eliminated unused software
  - Replaced software where appropriate with lower cost equivalents
  - Finding and fixing inefficient application code
COBOL Cost Containment

- At one point in time we had 4 different COBOL compiler versions
- Over time we eliminated 2 old versions
- We started upgrading the current version of COBOL and eliminating its predecessor
- We still have some 24 bit COBOL which we are slowly upgrading as these programs get upgrades
COBOL Issues

- The COBOL compilers were licensed for all machines
- Some programs got tested with one version of COBOL and placed in production with another
- Mismatched compile options between test and production
Solving the Problems

- A co-worker developed a compile process for test and production to solve the compile version and options problems.
- We canceled the 24-bit COBOL license, all COBOL compiles must now convert to the current (31-bit) COBOL version (IBM COBOL V4 5655–S71).
- We were planning on a multi-job process routing all compiles to LPARs on one machine:
  - IDMS and DB2 binds
  - Reduce licenses to one machine, reducing cost.
COBOL Compiles on the Cloud

- We found a provider offering COBOL compiles on the cloud, Cloud Compiling
  - Cloud compile is the ability to shift the compiles off your mainframe to the vendor’s data center
- Eliminates the need for multi-job compile processes
- Our contract negotiation provided lower cost than our IBM VWLC costs
  - We pay an attractive percentage of the IBM VWLC cost for COBOL compiles
  - We audit the cloud compile usage and VWLC cost with LCS from I/S Management Strategies, Ltd.
We kept our license for IBM COBOL V4 (5655–S71) so we could do compiles in–house if the network connections or service are not available.

The COBOL V4 no 89 parameter in SCRT was set to *NONE
  – * IBM ENTERPRISE COBOL FOR Z/OS V4
  – 5655–S71=*NONE

We pay the 0–3 MSU charge every month to keep the compiler available.
By using customized software (named exactly the same as the compiler), the following occur:

◦ The compile is intercepted at the compile job step
◦ The cloud compile software dynamically builds FTP files to send the expanded source code to the vendor site
◦ Once the source FTP (USS task) is successfully completed, the in-house compile job is placed into a wait state.
◦ The vendor site compiles the source
◦ A FTP of the output of the vendor compile back to the original compile job is invoked
◦ Once the FTP is successfully completed, the in-house compile job continues through the rest of its steps

Note, all compile options must be specified to guarantee a proper compile
The Implementation Path for Cloud Compile

- The customized software was initially implemented into a “TEST” library
- A wide variety of tests were submitted using the test library (CICS, IDMS, DB2, Batch, and combinations of software)
- When testing was complete, the standard compilation tool was modified to use the new library
- When it proved to be fully functional, the new test library members were copied into the production libraries for all compiles to use
There are three identifiable output files that come out of Cloud Compile that do not come out of in-house compiles.

- **TCCPRINT** contains the following:
  - TCC0001I The Cloud Compiler CC 5655–S71 V3R1.3 starting at 12:43:02 on 12/06/2012 (Build 208011818)
  - TCC0042I Cloud Compiler parameter file is CLOUDCMP.CNTL(SVCPARMS)
  - TCC0057I Starting processing on Facility FNTS at 12:43:02
  - TCC0035I Cloud Compiler terminating at 12:43:09, return code 0

- **TCCFTP** contains all information about the FTP processes to and from the cloud compile vendor’s location.

- **TCCSYSMS** contains the JES2 Job Log from the vendor’s location for the compile.

The cloud compile program creates USS tasks to perform the FTP processes with the proper parent child relationships.
CPU utilization in the compile step is 2 to 3 times higher with cloud compile due to the dynamic FTP steps. While it is higher, it is still a miniscule amount.

The wall clock time is about 2 to 3 times longer in the compile step.

Because of the USS FTP tasks, we discovered COBOL compiles that we didn’t know about, occurring in TSO.

Having multiple CEC’s and LPAR’s several of which are licensed to use the compilers, is a significant software licensing cost. By eliminating the need for these licenses, a significant savings is achieved.

So far we have had no legal, security, or integrity issues with sending our COBOL programs to the cloud for compiling.

THE JUICE IS worth the squeeze!
A Quick Aside

- We also looked at our C++ compiler
- It was licensed on one machine only
- We found out we had done 3 C++ compiles in the previous 12 months
  - 2 users
  - 3 tests
- We cancelled the license for the compiler
- Since that time we have had 1 time we needed it to compile a vendor exit, it was rewritten in assembler
Questions?
Thanks!

- Rick Howlett, Humana – The Compile process, implementation and the significant portions of this presentation
- Tom Wilson, Humana – Installation of the Cloud Compile software
- Al Sherkow, I/S Management Strategies – LCS enhancements to audit/report on cloud compile usage
Who provides Cloud Compile?

Budd J Rutter II | Founder

Phone: 651-216-2222
Toll Free: 877-245-4322
Email: budd.rutter@CloudCompiling.com
www.CloudCompiling.com